

Absolvování individuální odborné praxe
Individual Professional Practise in the Company

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2009

.....

Rád bych na tomto místě poděkoval firmě NetDevelo za možnost absolvování odborné praxe, Petrovi Hrubému a Michalovi Pindákovi, za odborné vedení a předané zkušenosti. Dále bych rád poděkoval své přítelkyni Monice, rodině a přátelům za jejich podporu a trpělivost se mnou.

Abstrakt

Cílem této bakalářské práce je podrobná zpráva o absolvování odborné praxe. V úvodní kapitole se nachází bližší informace o firmě mnou navštěvované. Popis obsahuje specifikaci firmy, poskytované služby a základní informace o hlavním produktu firmy. Dále v této kapitole popisuji mé pracovní zařazení. Druhá kapitola je rozdělena na dvě podkapitoly, každá z pohledu jiného pracovního zařazení. V každé z těchto kapitol je nejprve popsáno, co je pracovní náplní pracovníka na dané pozici, dále zadané úkoly a postupy při jejich řešení. Závěr každé z kapitol obsahuje zhodnocení dané pracovní pozice a také popis scházejících, resp. získaných zkušeností během vykonávání odborné praxe. V závěrečné kapitole hodnotím celou odbornou praxi a její přínos pro mě.

Klíčová slova: e-shop, JAVA, bakalářská práce, testování, napojení na EIS

Abstract

Objective of this bachelor thesis is detailed report about completion practical experience. In first chapter are more detailed informations about company. Content of this chapter is specifications of company, provision of services and basic information about leading product. In the end of this chapter i describe my official position. Second chapter is devided on two subchapters. In both subchapters is job description, my jobs and procedures for resolving. At the end of both subchapters is assessment of my official position and description of missing and background of experience. In finally chapter is description my practical experience and benefits for me.

Keywords: e-shop, JAVA, bachelor thesis, testing, connection on EIS

Seznam použitých zkratk a symbolů

B2B	– Business to Business
B2C	– Business to Customer
CSV	– Comma-separated values
EIS	– Ekonomický Informační Systém
SAX	– Simple API for XML
SEO	– Search Engine Optimization
URL	– Uniform Resource Locator
XML	– eXtensible Markup Language
backend	– část e-shopu přístupná jen pro administrátora, tzv. administrace e-shopu
design	– návrh struktury a vzhledu (aplikace,e-shopu)
e-shop	– elektronický obchod
frontend	– část e-shopu přístupná pro návštěvníka
java	– oběktově orientovaný programovací jazyk
swing	– knihovna tříd a komponent pro tvorbu grafického uživatelského rozhraní v Javě

Obsah

1	Úvod	2
2	Obecné informace	3
2.1	Informace o firmě	3
2.2	Pracovní zařazení	3
3	Tester nebo Programátor?	4
3.1	Na pozici testera	4
3.2	Na pozici programátora	6
4	Závěr	11
5	Literatura	12

1 Úvod

S velkým problémem při výběru zaměstnání se potýká většina absolventů vysokých škol. Důvodem jsou požadavky firem na uchazeče o místo. Většina firem požaduje od uchazečů o místo odbornou praxi. Tady nastává problém pro absolventy, protože při těchto podmínkách není lehké získat praxi v oboru. Musí pak často volit nepřímocárý a zdlouhavý postup při budování kariéry.

Absolvování odborné praxe v rámci bakalářské práce proto vítám a volím s vidinou získání zkušeností v oboru. Součastně doufám ve zvýšení svých referencí při výběru budoucího zaměstnání.

Kapitola 2 obsahuje bližší informace o firmě, ve které jsem absolvoval odbornou praxi a dále také mé pracovní zařazení. V kapitole 3 popisuji pozici, na které jsem pracoval, rozbor zadaných úkolů a popisů postupů při jejich řešení, dále také získané a chybějící zkušenosti. V závěru kapitoly hodnotím dosažené výsledky. Kapitola 4 obsahuje hodnocení celé odborné praxe a vyvození důsledků pro mě samotného.

2 Obecné informace

2.1 Informace o firmě

Společnost Netdevelo s.r.o. se zabývá vývojem profesionálních internetových aplikací. Doménou společnosti je internetový obchod a prodej ve všech jeho podobách. Pilotní produkt je internetový obchod ShopSys®. Související oblastí působnosti společnosti je tvorba internetových prezentací a síťových aplikací. Mezi poskytované služby patří :

- optimalizace stránek pro lepší umístění ve vyhledávačích (SEO analýza, SEO optimalizace),
- poradenství ve strukturování informací pro jejich správné prezentování na internetu,
- propojení e-shopu s ekonomickými programy,
- tvorba grafiky, reklamy a bannerů,
- provoz serverů a webhosting.

Firma poskytuje svůj pilotní produkt ShopSys® ve třech variantách podle individuálních požadavků zákazníka. Varianta LITE byla vyvinuta zejména pro cílovou skupinu zákazníků, kteří neplánují obchodovat s větším množstvím položek. Je to nejzákladnější varianta aplikace ShopSys®. Varianta PROFI je určena středně velkým až velkým e-shopům s neomezeným počtem položek a přijímaných objednávek a lze na ní vystavět B2B i B2C řešení. Tato varianta dále připojuje možnost realizovat propojení (pomocí aplikaci JShopSys viz. níže) e-shopu s podnikovým, ekonomickým či účetním systémem zákazníka. Umožňuje také individuální grafický design. Varianta ENTERPRISE navíc podporuje provoz na více různých doménách, multiměnovost ¹ a také multijazyčnost. Dále je také věnována zvýšená pozornost SEO optimalizaci [4].

JShopSys je také jedním z projektů firmy a jedná se o aplikaci realizující přenos dat mezi e-shopem zákazníka a jeho ekonomickým systémem. Dále také export dat zpět do e-shopu (podrobný popis funkčnosti aplikace v kapitole 3.2).

2.2 Pracovní zařazení

Do firmy jsem byl přijat na pozici testera webové aplikace ShopSys®. Na základě mého požadavku při přijetí jsem byl později dosazen do pozice programátora aplikace JShopSys.

Během odborné praxe jsem tedy pracoval na dvou různých pracovních pozicích. Z těchto důvodů je následující text strukturován do dvou oddělených kapitol. V každé kapitole budu referovat o úkolech, které mi byly zadány, jaké řešení jsem zvolil a jaké dovednosti jsem získal při jejich řešení.

¹ multiměnovost - vlastnost e-shopu, umožňující nastavit u zboží pro každou měnu odlišnou cenu.

3 Tester nebo Programátor?

3.1 Na pozici testera

Na tuto pozici je nejlepším kandidátem člověk neprogramátor, tedy člověk, který neovládá programovací jazyky, ale je schopen dobře zhodnotit grafický návrh aplikace (případně navrhnout zlepšení) a její funkčnost. Práce testera spočívá v testování všech možných scénářů, které mohou nastat při používání dané aplikace a nalezené nesrovnalosti dále zpracovat. V případě testera to znamená zadání chyby do systému pro sledování chyb, jako je například Bugzilla, nebo předáním úkolu přímo programátorovi. Tester je posledním článkem ve vývoji projektu. Po jeho schválení se projekt předává zákazníkovi. Pokud i pak obsahuje předaný projekt chybu, zodpovědnou osobou se stává tester nikoliv programátor.

V mém případě bylo obsahem práce testování jak administrace (backend), tedy části, kterou obsluhuje budoucí majitel e-shopu, tak testování e-shopu z pohledu potenciálního návštěvníka tzv. frontend.

3.1.1 Zadané úkoly a zvolené postupy

3.1.1.1 První důležitý úkol a zároveň nutnost pro každého nově příchozího pracovníka je seznámení se s aplikací ShopSys®. Frontend aplikace obsahuje jako každý profesionální e-shop standardní prvky (hlavička s logem společnosti, hlavní menu, menu po stranách s navigací, přihlašovací formulář, ankety apod.), které nepotřebují další komentář. Důležitější vlastnosti nabízí administrace e-shopu. Pro testera je důležité, aby znal všechny funkce a mohl je správně testovat.

Po seznámení s aplikací nebrání nic tomu začít testovat již spuštěný e-shop (spuštěným e-shopem je myšlena funkční aplikace, běžící na doméně zákazníka). Při prvním testování jsem nebyl jediným testerem daného e-shopu, ale i tak jsem si měl možnost vyzkoušet, jak testování v praxi probíhá. V dalším odstavci bych obecně popsal, co všechno by měl tester při testování dodržet a co všechno patří mezi jeho pracovní povinnosti.

První co musí tester zkontrolovat je, zda-li byla zákazníkovi nasazena správná verze e-shopu shodující se s aktuálně vyvinutou verzí. Pak následuje testování grafiky a funkčnosti e-shopu. Po grafické stránce musí vzhled e-shopu vyhovovat grafickému návrhu, který zákazník odsouhlasil při podpisu smlouvy. Co se funkčnosti týká, volí si zákazník individuální vlastnosti (dále jen moduly) e-shopu, které bude používat. Všechny moduly, které firma nabízí, musí tester znát a podle toho je testovat. Pro přiblížení uvedu krátký seznam několika modulů a jejich popis:

- multidoménovost - provozování více e-shopů (na různých doménách, s různým vzhledem) se společnou administrací,
- multijazyčnost - doméně e-shopu odpovídá také jazyk,
- multiménovost - podle jazyka e-shopu příslušná měna,

- napojení na EIS - propojení e-shopu s ekonomickým systémem (pomocí aplikace JShopSys),
- SEO optimalizace - zaručení validity kódu, čistých URL adres atd.

Po ukončení testování musí tester všechny nalezené chyby popsat do systému pro zaznamenávání chyb. Nadřízený pracovník rozdělí práci programátorům. Na opravu nalezených chyb musí tester dohlédnout a o průběhu oprav opět informovat nadřízeného pracovníka.

3.1.1.2 Příklad spolupráce a koordinace uvedu v následujícím odstavci. Úkolem bylo otestovat správnou funkčnost objednávání zboží. Vytvořil jsem tedy v administraci e-shopu několik druhů zboží. Kolega poté ve frontendu e-shopu vytvořil objednávku obsahující mnou vytvořené zboží. Po odeslání objednávky bylo nutné tuto objednávku zkontrolovat v administraci. Je nutné ověřit, zda-li objednávka obsahuje všechny povinné údaje uživatelem vyplněné, obsahuje-li zboží uživatelem vybrané apod. Důležitá je také kontrola všech výpočtů při stanovování celkové ceny (zaokrouhlování, výpočet daně, atd.). Po ověření správnosti jsem objednávku označil jako vyřízenou a kolega ještě zkontroval, jestli se tato změna projeví i na frontendu.

3.1.2 Získané a chybějící dovednosti během odborné praxe

Za hlavní přínos této části praxe považuji získanou zkušenost. Díky této možnosti jsem se dostal k testování běžící aplikace a s tím spojenému řešení v reálném čase. Novou zkušeností je pro mě také nutnost komunikace s kolegy a předávání si informací, získaných při testování. Doposud jsem byl zvyklý otestovat si sám aplikaci, kterou jsem implementoval. V týmu již o dvou lidech je potřebné synchronizovat úsilí členů tak, aby bylo efektivní (viz. 3.1.1.2). Tyto zkušenosti pro mě jsou určitě hlavním přínosem.

Dovednosti, které mně chyběly při testování e-shopu, byly spíše komunikativního rázu, týmové spolupráce než odborné. Naučil jsem se během krátké doby rychlé a efektivní komunikaci mezi kolegy v mluvené i psané formě.

3.1.3 Dosažené výsledky jako Tester

Hlavním přínosem této části praxe je získaná zkušenost. Osvojil jsem si všechny povinnosti testera a měl možnost si v praxi vyzkoušet, co tato práce obnáší. Testováním aplikace ShopSys® jsem porozuměl logice funkčnosti a to považuji za výhodné, zejména pro další setrvání ve firmě.

3.2 Na pozici programátora

Jedná se o pozici programátora v programovacím jazyce JAVA. Programátor na této pozici má za úkol přizpůsobit aplikaci JShopSys zákazníkovi podle jeho požadavků. Mezi nejčastější úpravy patří uzpůsobení aplikace podle ekonomického systému, který zákazník používá, a dále jaké data požaduje zákazník přenášet do svého ekonomického systému. V následující definici podrobněji popíšu základní funkčnost aplikace JShopSys.

Aplikace JShopSys (obr.1) je implementována v programovacím jazyce JAVA [1, 2]. Pro tvorbu grafického uživatelského rozhraní je použita knihovna Swing [3]. Aplikace běží na lokálním stroji zákazníka a umožňuje nastavit, jaké akce se mají provádět. Mezi tyto akce patří například:

- import ² a export zboží ³
- import a export objednávek
- import a export faktur

Tyto akce se provedou buď ihned po zmáčknutí tlačítka "provést akci ihned" nebo po uběhnutí časového intervalu nastaveného uživatelem.

Další důležitou informací je, v jakých formátech jsou data přenášena. Všechny JShopSys e-shopy generují např. při vzniklé objednávce XML soubor a ten je importován do ekonomického systému uživatele. Ekonomické systémy generují např. při vzniku faktury také XML sooubor a ten je exportován zpět do e-shopu.

Při exportu zboží provádí aplikace JShopSys parsování XML souborů. Kontroluje pouze element status, ve kterém se nachází informace o případné chybě. Pokud nějaká taková chyba existuje, je vypsáno informativní hlášení uživateli na obrazovku. Záznam o chybě je také zapsán do logovacího souboru.

3.2.1 Zadané úkoly a zvolené postupy

3.2.1.1 Prvním úkolem ryze seznamovacího typu bylo prozkoumat samotnou aplikaci JShopSys, vyzkoušet si její funkčnost a možnosti. Vedlejším úkolem bylo oddělit všechny napevno zadané cesty k externím souborům (konfigurační soubory, lokalizační soubory apod.) mimo funkční části programu.

Řešení spočívalo ve vytvoření třídy obsahující statické proměnné ukazující na konkrétní soubory (výpis 1). Takto připravené proměnné není problém použít kdekoliv v kódu a při případné změně některé z cest se tato změna pouze na jednom místě.

²import - přenos dat z e-shopu do ekonomického systému.

³export - přenos dat z ekonomického systému zpět do e-shopu.

```

package configuration;

public class PathFiles {

    /**
     * cesty k lokálním souborům
     */
    public static final String settingsFile = "config/settings.ini";
    public static final String lokalizacniSoubor = "config/lokalizace/";
}

.
.
.

import configuration.PathFiles;

public class UkazkovaTrida {

    String jazyk = "cs";
    File soubor = new File(PathFiles.lokalizacniSoubor + jazyk + ".lng");
}

```

Výpis 1: ukázka použití třídy PathFiles.

3.2.1.2 První skutečný zásah do kódu znamenal povýšení aplikace na vícejazyčnou. Takové bylo zadání mého dalšího úkolu. To znamená, aby si uživatel mohl jednoduše zvolit jazyk aplikace podle potřeby. Vedlejší efekt tvorby vícejazyčné aplikace spočíval ve shromáždění všech řetězců použitých v aplikaci do jednoho souboru, resp. dvou souborů s českým a anglickým překladem. Při potřebě změnit jakýkoliv řetězec používaný v aplikaci stačí změnu provést pouze v lokalizačních souborech.

Řešení se pokusím popsat po jednotlivých krocích dle průběhu programu. Předem upozorňuji, že všechny použité metody jsou statické a použití lokalizovaného řetězce kdekoli v kódu znamená zavolat metodu `getWord`, které předáme argumentem konstantu s hodnotou kýženého slova. Metoda `getWord` tedy již vrací lokalizovaný řetězec. Při prvním volání metody `getWord` proběhne inicializace, při dalších voláních metody již bude inicializace přeskočena. Metoda inicializace načte z konfiguračního souboru aplikace jazyk, který je defaultně nastaven a podle této hodnoty otevře stream k souboru daného jména. Dále je přečten obsah celého lokalizačního souboru najednou do kolekce typu `Vector`. Důvodem přečtení celého souboru najednou je to, že čtení ze souboru je nejkritičtější (nejpomalejší) částí programu. Proto je soubor přečten na jednou a není již potřeba opakovaně číst ze souboru. V tuto chvíli je ukončena inicializace. V poslední fázi je programem zavolána metoda, která prohledává kolekci naplněnou hodnotami z lokalizačního souboru a při úspěchu předá metodě `getWord` nalezený řetězec. Při neúspěchu je vrácena hodnota konstanty, která byla předána metodě `getWord` ohraničená znakem `'?'`.

3.2.1.3 Induviduální úpravy aplikace byly po většinu času náplní mé práce. Úkoly, které jsem popisoval doposud, by se daly označit s trochou nadsázky jako vývoj aplikace. Realita je bohužel taková, že na vývoj aplikace nezbývá mnoho času a každodenní náplní programátora se stává uspokojování potřeb zákazníka. V mém případě tomu samozřejmě nebylo jinak. Náplní mé práce bylo po většinu času plnění požadavků na různé malé úpravy a vylepšení aplikace přesně podle individuálních požadavků zákazníka. Pokusím se tedy popsat o jaké úpravy se jedná.

Mezi nejčastější požadavky zákazníků patří upravení vzhledu aplikace ve smyslu změny pořadí záložek v aplikaci, odstranění záložky pokud ji zákazník vůbec nepoužívá apod. Problém může nastat také při importu nebo exportu dat. Důvodem takovéto chyby bývá špatná struktura XML dokumentu a v takovém případě je potřeba se ze zákazníkem domluvit, jaká bude přenášovaná struktura dat. Chyba může také nastat, pokud zákazník nastaví příliš krátký interval pro import dat a díky pomalému internetovému připojení nebo velikosti XML souborů dochází k pádu aplikace. Tato chyba je samozřejmě spíše problémem aplikace JShopSys, která by měla takovou situaci řešit čekáním na nedokončený přenos.

3.2.1.4 Splněním dalšího úkolu jsem opět zasáhl do "vývoje" aplikace. Požadavkem bylo kontrolovat všechny XML soubory objednávek importované z e-shopu do EIS (výpis 2) a případné chyby zobrazovat uživateli výpisem na obrazovku, ale také zaznamenávat do logovacího souboru pro případ zpětné kontroly.

K řešení jsem použil SAX parser a to z důvodu předem neznámé velikosti XML souboru. Při použití parseru DOM by mohla aplikace u většího XML souboru vyčerpat všeskerou operační paměť počítače. Výhodou SAX parseru je, že jeho paměťová náročnost je nesrovnatelně menší než u DOM parseru. SAX parser může při čtení XML souboru reagovat jen na určité události. Výsledné chování kódu tedy je:

- parsování XML souboru,
- pomocí třídy `StringBuffer` záznam obsahu důležitých elementů,
- výpis informací uživateli na obrazovku, jedná-li se o závažnou chybu,
- v případě že parsovaný XML soubor je v pořádku, zápis informací pouze do logovacího souboru.

Ještě bych dodal, že se vždy neprovádí parsování celého souboru. Každý XML dokument vygenerovaný e-shopem obsahuje jako jeden z prvních elementů element "Status" a podle hodnoty tohoto elementu je rozhodnuto, zda-li se bude parsovat nebo je objednávka v pořádku a pokračuje se další objednávkou.

```

public class Parser{
    public static void run(String soubor){
        try{
            SAXParserFactory spf = SAXParserFactory.newInstance();
            SAXParser saxlevel1 = spf.newSAXParser();
            XMLReader parser = saxlevel1.getXMLReader();
            parser.setErrorHandler(new ChybyZjisteneParserem());

            // volani konkretniho handleru
            ObjednavkyHandler m = new ObjednavkyHandler();
            parser.setContentHandler(m);
            parser.parse(soubor);
        }
        catch (SAXException e){
            //implementace vyjimky
        }
        // dalsi implementace
    }
}

public class ObjednavkyHandler extends DefaultHandler{
    public void startElement(String uri, String localName, String qName, Attributes atts){
        if (qName.equals(SEZNAMOBJPRIJ) == true){
            SeznamObjPrij = true;
        }
    }

    public void endElement(String uri, String localName,String qName){
        if (qName.equals(SEZNAMOBJPRIJ) == true){
            SeznamObjPrij = false;
        }
    }

    public void characters(char[] ch, int start, int length){
        if (SeznamObjPrij == true){
            buffer.append(ch, start, length);
        }
    }
}

```

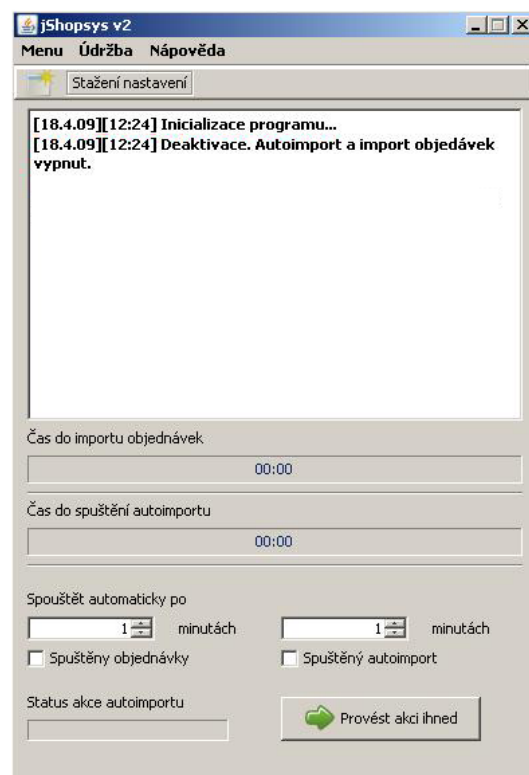
Výpis 2: ukázka implementace SAX parseru.

3.2.2 Získané a chybějící dovednosti během odborné praxe

Během této části praxe jsem získal mnoho užitečných zkušeností. Předně to je práce s vícevláknovou aplikací a její vývoj. Dále práce s XML soubory, ať už jejich čtení pomocí parserů SAX a DOM nebo jejich vytváření. Za důležité také považuji zkušenosti získané v komunikaci se zákazníkem.

3.2.3 Dosažené výsledky jako Programátor

Za dosažené výsledky bych označil všechny úkoly uvedené výše. At' už to je lokalizace aplikace JShopSys, nebo SAX parser, který jsem implementoval. Za výsledek se dají také považovat i všechny individuální malé upravy, které jsem udělal podle přání zákazníků.



Obrázek 1: Aplikace JShopSys

4 Závěr

Nejprve se několika řádky zmíním o samotné odborné praxi. Považuji za velkou výhodu možnost výběru individuální odborné praxe jako variantu ukončení bakalářského studia. Student má možnost vybrat si zaměstnání, které by chtěl v budoucnu vykonávat a může si vyzkoušet, jestli je tato práce to, co očekával. Pokud se student rozhodne zůstat i dále, firma přijímá zaměstnance, kterého si sama vychovala. V tom vidím obrovskou výhodu pro obě strany. Firma nemusí přijímat zaměstnance na základě životopisů, ale podle doposud odváděných výkonů studenta. Na druhou stranu student už dobře zná podmínky uvnitř firmy a ví, co ho čeká. V dnešní době, kdy každá firma požaduje od uchazečů o místo praxi v oboru, se jeví tato varianta bakalářské práce víc než výhodná.

Během mé odborné praxe jsem měl možnost pracovat jako tester internetových elektronických obchodů a v druhé části jako programátor aplikace v programovacím jazyce Java.

Práce na pozici testera nebyla důvodem mého příchodu do firmy, ale znamenala pro mě spíše nezbytnou podmínku pro další postup. Na druhou stranu i při této práci jsem získal mnoho užitečných zkušeností. Mnohem důležitější jsou pro mě získané zkušenosti programátora. Díky dobrému vedení zkušených programátorů jsem se přiučil novým věcem, osvojil jsem si používané standardy.

Odbornou praxi hodnotím jako vysoce přínosnou a věřím, že nabyté zkušenosti uplatním při mé další odborné činnosti.

5 Literatura

- [1] HEROUT, Pavel. *Java Bohatství knihoven*. II.upravené a rozšířené vydání. České Budějovice: KOPP, 2006, ISBN 80-7232-288-5
- [2] HEROUT, Pavel. *Java a XML*. České Budějovice: KOPP, 2007, ISBN 978-80-7232-307-4
- [3] *informace o knihovně swing*.
URL: <<http://java.sun.com/docs/books/tutorial/uiswing/components/index.html>>
[cit. 2009-04-27]
- [4] *Netdevelo s.r.o.*
URL: <<http://www.netdevelo.cz>> [cit. 2009-04-19]